Extracting Addresses From News Reports Using Conditional Random Fields

Xiaoqian Liu and **Donald Brown**
Department of Systems and Information Engineering, University of Virginia
{xl3ma,deb}@virginia.edu

Abstract

Spatial analysis in many fields requires effective address extraction from text reports. This problem is of particular importance in social science where news reports contain information about socially relevant incidents. To address this problem this paper presents an approach to automatically extract street addresses from news reports by combining sequential labeling with semi-supervised learning. Previous work on address extraction focused on web pages where addresses are separated from other text; however news reports contain addresses embedded in text. Hence, the need for the approach described here. Experimental results from the application of the approach to actual news reports show performance close to that achieved for web pages.

1 Introduction

Address information is widely used for spatial analysis in many applications. Many tools exist for extracting addresses from emails, web pages, and other files with special tagging. For example, address extraction from web pages uses specific div tags or address blocks which are completely independent from other text. However, these methods do not apply to address extraction from within free text, such as, news reports..

Address extraction from news reports has several challenges. First, there is no special formatting; so, the address maybe anywhere in the text. Second, there are no labeled news report datasets for training and testing. Creating these data sets requires human supervision. Finally, a single address may have various forms in the news reports. For example, the author may neglect the street number, use street abbreviations instead of full names or misspell the names. This last problem also exists for web pages (see (Yu, 2007)).

This paper describes the use of word-level features to improve predictability using measures such as the minimum word distance to the keywords of the target of interest. Additionally, the approach here shows the use of gradient boosting, Principal Component Analysis and Conditional Random Fields (CRF) for labeling with both supervised and semi-supervised learning. The semi-supervised learning algorithm is based on the work of Liao and Veeramachaneni (2009). The results in this paper show an experimental comparison of these methods with news reports on sexual violence obtained from the Washington Post and a University of Virginia student paper.

2 Related Work

Name Entity Recognition (NER) works to retrieve and identify information units, such as person, location, and organization (Ratinov and Roth, 2009). Location can be subdivided into addresses such as city and state (Nadeau, 2007). Under recent conventions, names are marked-up with "Enamex" tags. Since "Enamex" tagging concerns more varied patterns than addresses, there are no direct applications of existing NER systems to address extraction from news.

Pattern-based methods for address extraction mainly use Gazetteers and Regular Expressions with

heuristic rules. A gazetteer is a geographical dictionary which can provide indexed spatial information such as states, cities and streets. Using geographic specific indices with a Gazetteer lookup achieved an F1 score of 75.3% which outperformed regular expression methods (Yu, 2007).

Another popular approach for address extraction is machine learning. Recent research applied decision tree models as well as the pattern based methods described above (Yu, 2007). This approach first tokenized a web page document and categorized tokens into one of four classes, namely Start, Middle, End and Others. After the classification step, Yu's method retrieved addresses based on the label sequence. According to the evaluation results, the decision tree with a regular expressions had the best performance: precision value of 95.2% and recall of 81.1%. Building on this work, Chang and Li created MapMarker for extracting postal addresses and associated information (Chang and Li, 2010). They exploited the CRF model for classification and obtained prediction accuracy with an F1 of 91% .

3 Address Extraction Methodology

The basic address extraction methodology in this paper has five parts: 1. Tokenization; 2. Generation of n-grams; 3. Labeling; 4. Sampling; 5. Classification; and 6. Address retrieval. The first step, tokenization, splits the document into sentences and then uses regular expressions to tokenize each sentence. Next n-grams are created by concatenating tokens in a context window size of $\frac{(n-1)}{2}$. The elements of the n-gram are labeled: beginning of the address (B); inside of the address (I); end of the address (E); and outside of the address (O) (Yu, 2007; Chang and Li, 2010). The class label for each n-gram is determined by its central word.Since the number of "O" labels overwhelms the combined number of "B", "I", and "E" labels by as many as 4 orders of magnitude, our approach subsamples the "O" labels. Testing showed that a ratio of 2:1 for "O" versus "B"+ "I" + "E" labels worked well.

The next step in the approach is classification into one of seven classes: location: person: organization: money: percent: date; and time. The classes other than location provide a foundation for future work. The classification algorithms use 17 word-level features (e.g. street suffixes, ordinal directions, regular expression pattern matching, and the Stanford NER tagger (Finkel et al., 2005)). The approach described here uses three algorithms: Gradient Boosting (GB); Gradient Boosting with Principal Component Analysis (GB-PCA); and Conditional Random Fields (CRF).

For GB the implementation is XGBoost (Chen and Guestrin, 2016), which is a scalable and computationally-efficient implementation of GB. Tree boosting has the following regularized learning objective.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{1}$$

Starting with a data set which contains n examples and m features, the boosting model uses the summation of k feature functions, to predict the output. In this equation, $F = \{f(x) = w_{q(x)}\}(q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$represents the space of regression trees. q is a tree structure composed of a set of decision rules. It maps an example to its corresponding leaf index. Each f_k has its associated leaf weights w and a tree structure q.

Our address extraction methodology also uses GB with PCA (GB-PCA). PCA provides dimension reduction of the word level features where each principal component is a linear weighting of the original features. The resulting components are pairwise orthogonal and represent as much of the original variance in the data as possible.

CRF is an undirected graphical model that is widely used in Part-of-Speech Tagging, Name Entity Recognition and many other Natural Language Processing tasks. In CRF X, a random vector, is a collection of input sequences and Y is a random variable over label sequences (Sutton and McCallum, 2011). CRF is then a random field globally conditioned on X. Its formula is as following:

$$\begin{aligned} p(y|x) &= \frac{1}{Z(c)} \exp(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)) \\ Z(x) &= \sum_{y} \exp(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)) \end{aligned} \tag{2}$$

In equation (2), Z is the normalization factor, λ_k is an array of weight parameters and f_k represents real-valued feature function which can be in various forms (e.g. prefix of xt and features of surrounding words). Parameter estimation in CRF is solved by penalized maximum likelihood (Sutton and McCallum, 2011).

The classification algorithms require a training set of correctly labeled exemplars. This is a manual, time consuming task that results in only small data sets for training and testing. Semi-supervised learning increases the size of these sets by automatically labeling unlabeled data. Our approach uses Algorithm 1 due to Liao and Veeramachaneni (2009)

Algorithm 1 Semi-Supervised Learning

Given:
L - a small set of labeled training data
U - unlabeled data
for $k \in K$ iterations **do**
  Train a classifier C_k based on L
  Extract new data D based on C_k
  Add D to L

The final step in the address extraction methodology is to retrieve addresses. Algorithm 2 accomplishes this task.

Algorithm 2 Address Extraction

for each predicted output **do**
  if $\hat{y}_t = 3$ and $\hat{y}_{t-1} = 3$ **then**
    if the ending word or second word is valid **then**
      output the address
    else
      if a word is not a punctuation and distinct from the last word **then**:
        output the address
      create a new address candidate
return b

4 Experimental Results

Evaluation. Data for evaluation come from sexual assaults reported in the Washington Post and the Cavailier Daily (a student-operated daily newspaper in Charlottesville, VA, USA) from 1990 to 2015 as shown in Table 1. Sexual assaults were used to filter the news reports since this is a topic of interest to social scientists and frequently has address information. The unlabeled dataset is a combination of both news sources and is used for semi-supervised learning (Algorithm 1). The last row in Table 1 shows 213 reports of sexual assaults that appeared over the same period from other news reporting sources in the Charlottesville, VA, USA area. These manually labeled data provide the test set for the semi-supervised approach.

Table 1: News Reports Data Overview

Source	Count
Washington Post	374 Documents, 237 Addresses
Cavalier Daily (University Wire)	731 Documents, 132 Addresses
Unlabeled Data	998 Documents
All news reports	213 Documents

Performance measures for the approach consist of precision, recall and F1-score on the classification on a randomly chosen 30% of the data. The modeling parameters are tuned with 10-fold cross-validation of the training sets. Incomplete identifications of spans (i.e. classes) are neglected and false positive errors are not penalized.

Results. Table 2 shows the results on the Washington Post reports. These results show that the approach with all classifiers does well on "B" and "I" but not on "E". These results are comparable to those obtained on web page data (Chang and Li, 2010).

Table 2: Results on the Washington Post data set

Class	Model	Precision	Recall	F1-Score
B	XGBoost	0.91	0.90	0.91
	XGBoost+PCA	0.91	0.91	0.91
	CRF	0.95	0.87	0.91
I	XGBoost	0.95	0.87	0.91
	XGBoost+PCA	0.95	0.85	0.90
	CRF	0.98	0.88	0.93
E	XGBoost	0.71	0.62	0.67
	XGBoost+PCA	0.64	0.44	0.52
	CRF	0.53	0.50	0.52

The results for the student news are in Table 3. For "B" GB does better than CRF but worse than the results from the Washington Post, while CRF does better for "I" and "E". For this source performance is better for "E" than the other two classes. This

highlights the difference in reporting styles for addresses between the two news sources and the need for flexible address extraction methods.

Table 3: Results on the Cavalier Daily data set

Class	Model	Precision	Recall	F1-Score
B	XGBoost	0.86	0.86	0.86
	XGBoost+PCA	0.87	0.92	0.89
	CRF	0.81	0.92	0.86
I	XGBoost	0.50	0.09	0.15
	XGBoost+PCA	0.80	0.36	0.50
	CRF	0.80	0.73	0.76
E	XGBoost	0.84	0.80	0.82
	XGBoost+PCA	0.85	0.73	0.79
	CRF	0.85	0.91	0.88

Finally Table 4 provides a comparison of the classifiers with and without semi-supervised learning. Only CRF was used with the semi-supervised algorithm. The table shows that semi-supervised learning does improve performance for "B" and "I", but not "E". Since, the training set derived primarily from Washington Post data, these results are consistent with the earlier results but also suggest that supervised learning can help improve accuracy as well as training time. In this test we also used a decision tree classifier as was done in the earlier work by Yu (2007). This provides a comparison with the earlier method and the results show better performance over that baseline.

Table 4: Results on 213 all news reports

Class	Model	Precision	Recall	F1-Score
B	Decision Tree	0.78	0.75	0.77
	XGBoost	**0.85**	0.77	0.80
	CRF	0.84	0.71	0.77
	CRF+Semi	0.82	**0.89**	**0.85**
I	Decision Tree	0.60	0.75	0.67
	XGBoost	0.63	0.79	0.70
	CRF	0.71	0.79	0.75
	CRF+Semi	**0.76**	**0.83**	**0.79**
E	Decision Tree	0.68	0.61	0.64
	XGBoost	0.83	0.68	0.75
	CRF	**0.87**	**0.87**	**0.87**
	CRF+Semi	0.77	0.64	0.70

Overall the predictive accuracy of each class, "B", "I", and "E" is related to the percentage of correctly extracted addresses. The address extraction algorithm (Algorithm 2) starts to collect a temporary address when a component of the address is detected. It stops when a non-address element is found. Intuitively, the beginning of the address (class "B") is the most important span since it is the initial condition of the algorithm. When the F1-score of class "B" is high, the algorithm can more easily identify the existence of an actual address. Otherwise, the algorithm is more likely to neglect the existence of an address and that will reduce the accuracy of extraction. When the F1-score of class "E" is high, then the algorithm correctly captures the ending component. Nevertheless, when class "E" and class "I" labels are mixed up, the algorithm still works since it terminates when a non-address element is visited.

5 Conclusion

The address extraction methodology in this paper targets news reports and builds on previous work for address extraction from web pages. To accomplish this more complex extraction, the approach here added components, such as, CRF and GB to a more extensive process for extraction. These extensions improve the performance and produce results comparable to those achieved for web pages. Overall, CRF has the most stable performance on both datasets. Additionally the use of semi-supervised learning can provide good results and reduce training time and expense.

Future work should build on the semi-supervised results to create a larger data set for training which can then be used against a wider range of test sets. The results also suggest that an ensemble of classifiers may provide additional lift in performance. This ensemble can be adapted and trained for each of the separate labels, "B", "I", and "E". Finally, the approach can be expanded with other techniques such as word embeddings, recurrent neural networks, and more advanced and robust semi-supervised algorithms such as the general framework for structured learning and semi-supervised CRF using the generalized expectation criteria (Ando and Zhang, 2005; Mann and McCallum, 2008).

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Chia-Hui Chang and Shu-Ying Li. 2010. Map marker: Extraction of postal addresses and associated information for general web pages. In *Proceedings 2010 IEEE/WIC/ACM Int. Conference on Web Intelligence and Intelligent Agent Technology*, pages 105–111.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Wenhui Liao and Sriharsha Veeramachaneni. 2009. A simple semi-supervised algorithm for named entity recognition. In *NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing.* Association for Computational Linguistics.

Gideon S. Mann and Andrew McCallum. 2008. *Generalized expectation criteria for semi-supervised learning of conditional random fields.*

David Nadeau. 2007. Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Charles Sutton and Andrew McCallum. 2011. An introduction to conditional random fields. *Machine Learning*, 4.

Zheyuan Yu. 2007. High accuracy postal address extraction from web pages.